Queueing Systems manuscript No.
(will be inserted by the editor)

A Mean Field Model for a Class of Garbage Collection Algorithms in Flash-based Solid State Drives

Benny Van Houdt

Abstract Garbage collection (GC) algorithms play a key role in reducing the write amplification in flash-based solid state drives, where the write amplification affects the lifespan and speed of the drive. This paper introduces a mean field model to assess the write amplification and the distribution of the number of valid pages per block for a class $\mathcal{C}$ of GC algorithms. Apart from the RANDOM GC algorithm, class $\mathcal{C}$ includes two novel GC algorithms: the d-CHOICES GC algorithm, that selects d blocks uniformly at random and erases the block containing the least number of valid pages among the d selected blocks, and the RANDOM++ GC algorithm, that repeatedly selects another block uniformly at random until it finds a block with a lower than average number of valid blocks.

Using simulation experiments we show that the proposed mean field model is highly accurate in predicting the write amplification (for drives with $N = 50,000$ blocks). We further show that the d-CHOICES GC algorithm has a write amplification close to that of the GREEDY GC algorithm even for small d values, e.g., $d = 10$, and offers a more attractive trade-off between its simplicity and its performance than the WINDOWED GC algorithm introduced and analyzed in earlier studies. The RANDOM++ algorithm is shown to be less effective as it is even inferior to the FIFO algorithm when the number of pages b per block is large (e.g., for $b \geq 64$).

1 Introduction

A mean field model for a class of garbage collection (GC) algorithms in flash-based solid state drives (SSDs) is introduced in this paper given that the workload on the drive consists of uniform random writes. Before introducing

B. Van Houdt
Department of Mathematics and Computer Science,
University of Antwerp - iMinds, Belgium
E-mail: benny.vanhoudt@ua.ac.be

the specifics of SSDs it is interesting to note that the evolution of such drives can be reformulated in terms of a balls and bins system consisting of ρbN balls and N bins that can each hold up to b balls, where $N \geq 1$ and $b \geq 1$ are integers and $\rho \in (0, 1)$ is a real number such that ρbN is an integer. At time epoch t, for $t \in \{0, 1, \ldots\}$, a bin is selected by the so-called GC algorithm. If the selected bin at time t contains j balls (where $0 \leq j \leq b$), a single ball is selected uniformly at random $b - j$ times among all the bins and moved to the selected bin between time t and $t + 1$. As will become apparent further on, our interest lies in finding the distribution of the number of balls in an arbitrary and the selected bin at time t as t tends to infinity. The class of GC algorithms considered in this paper essentially demands that the probability of selecting a specific bin should (in a smooth manner) only depend on the number of balls in the bin and on the fraction of bins that contain exactly j balls, for $0 \leq j \leq b$, see Section 4 for an exact definition. Examples of such GC algorithms include the Random GC algorithm, which selects a bin uniformly at random, the Random++ GC algorithm, which corresponds to selecting a bin uniformly at random among the bins containing at most $\lfloor \rho b \rfloor$ balls, and the d-Choices GC algorithm, which chooses d bins uniformly at random and selects the bin containing the least number of balls among the d selected bins.

To understand the analogy between the SSD operation and the abovementioned balls and bins system we start by discussing the SSD structure. Data on a NAND flash-based solid state drive (SSD) is organized in N blocks that each contain a fixed number of b pages, where a page is the smallest writable unit and b is a power of 2 in practice, e.g., $b = 64$. The size of a single page is typically 2 to 4 Kbyte and there can be as many as 128 pages per block. In order to write data on a page, it must first be in an *erase* state. Individual pages cannot be erased, only entire blocks can be erased. As it would be very time consuming to update pages by completely rewriting a block, out-of-place writes are performed on an SSD. Hence, when a page is updated, it is typically stored on a new location on the drive and page holding the old data is marked as *invalid*, while the page containing the new data is marked as *valid*. In other words, a single page can be in three different states: erase, valid or invalid and a ball in the balls and bins system corresponds to a page in the valid state, while the bins corresponds to the blocks on the SSD.

Ideally we only wish to perform erase operations on blocks that contain invalid pages only. However, the GC algorithm, responsible for selecting the block to be erased, will often select blocks that contain some valid pages (in fact, depending on the GC algorithm blocks containing invalid pages only may not exist). This implies that these valid pages need to be temporarily stored in memory before the block erase can take place, even though no external write operation is requested for these pages. These additional internal write operations give rise to what is known as the *write amplification*, it is the ratio of the total number of writes to the number of externally requested writes. Hence, the distribution of the number of balls in the selected bin determines the number of internally required writes and therefore determines the amount of write amplification.

The write amplification not only slows down the operation of the SSD, but it also affects its lifespan. More specifically, flash memory decays and becomes unstable after a certain number of write-erase cycles (e.g., as few as 10000 in some consumer SSDs [6, 9]), thus the higher the write amplification of an SSD the shorter its lifespan. To limit the write amplification, the total storage capacity (number of physical pages) on an SSD exceeds the user-visible capacity (number of logical pages), as this guarantees that a fraction of the pages is in the erase or invalid state. A commonly used measure for the amount of over-provisioning is the *spare factor* S_f, defined as one minus the ratio of the user-visible to the total storage capacity. In our balls and bins system $\rho = 1 - S_f$, meaning ρbN pages are in the valid state at all times, while $(1-\rho)bN$ pages are either in the erase or invalid state. As will become apparent in Section 2, the externally requested writes will correspond to the balls being moved to the selected bin.

In this paper we introduce a mean field model to assess the write amplification and the distribution of the number of valid pages per block for a class $\mathcal{C}$ of GC algorithms under uniform random writes by relying on the framework introduced in [4]. We show that the mean field model is in perfect agreement with simulation experiments and compare the performance of the d-Choices and Random++ GC algorithm with the Greedy [5, 7], FIFO [7, 16] and Windowed [10] algorithm. We observe that the d-Choices GC algorithm can achieve a write amplification close to that of the Greedy GC algorithm even for small d values, e.g., $d = 10$, and offers a more attractive trade-off between its simplicity and its performance than the Windowed GC algorithm. The Random++ algorithm on the other hand is inferior to the FIFO algorithm when the number of pages b is large, e.g., for $b \geq 64$.

The flash translation layer, responsible for mapping the logical pages to physical page numbers, considered in this and the above mentioned papers is a page-level map, meaning data can be written on any page and a direct map that translates the logical to physical page numbers is maintained in memory. A block-level map reduces the memory consumption, but increases the write amplification as logical pages can still be mapped to any block, but only to one page within this block (determined by the logical page number). Consumer SSDs typically rely on some form of hybrid mapping [11], where some of the blocks are block-mapped and others are page-mapped to reduce the write amplification of random writes. When a hybrid mapping is used, *merge* operations that create new page-mapped blocks also need to be performed by the GC algorithm.

The d-Choices algorithm has been studied extensively in a classic balls and bins, hashing and load balancing setting (e.g., [2, 15, 18]) and was also proposed as a GC algorithm for solid-state drives in [12], a paper that is being published concurrently. The latter paper also proposes a mean field model for uniform workloads, but the system operation differs significantly from ours, as the write operations do not appear to rely on a log-structure (while in our system all writes make use of the so-called write frontier, see Section 2).

Further, the spare factor does not appear to be a model parameter in case of the uniform workload model in [12], while it plays a key role in our model.

The paper is structured as follows. Section 2 states the main problem, while Section 3 gives an overview of the related work. The class of GC algorithms $\mathcal{C}$ studied in this paper is introduced in Section 4 and the corresponding mean field model is presented in Section 5. Analytical and numerical results for the Random, Random++ and d-Choices GC algorithm are presented in Section 6 and 7, respectively. Conclusions are drawn and future work is discussed in Section 8.

2 Problem statement

Consider a flash-based SSD consisting of N (physical) blocks that are each contain b pages. At any point in time there is a special block called the *write frontier*. Pages will be written sequentially to the write frontier, until it is full. Assume at some point in time that the first $f < b$ pages of the write frontier are in the valid or invalid state, while the last $b-f$ are in the erase state and a write operation takes place on a logical page that is physically stored on page k of block number n_1. This operation writes the new content to page $f+1$ of the write frontier, changes the state of page $f+1$ from erase to valid, and afterwards invalidates page k on block number n_1. Note, it is possible (though unlikely) that block number n_1 is in fact the write frontier itself (if $k \leq f$) and the write operation thus invalidates one of the first f pages of the write frontier.

When the write frontier becomes full, meaning the last of its pages in the erase state becomes valid, the GC algorithm creates a new write frontier as follows: it first selects a new block, say block number n_2, copies all the valid pages of block n_2 to the random-access memory (RAM), erases block number n_2 and copies the valid pages back from RAM to block n_2[1]. In our balls and bins system time epoch t corresponds to the t-th time that the GC algorithm is executed and the write frontier corresponds to the selected bin. The write operations in between two executions of the GC algorithm invalidate a page in some block and validate a block in the write frontier, as such they correspond to moving a ball from some bin n_1 to the selected bin.

If the GC algorithm selected a block containing $j < b$ valid pages, $b-j$ additional writes can be performed before the execution of the GC algorithm. This implies that b internal write operations took place in between two executions of the GC algorithm, while only $b-j$ external write operations were performed. In this case the *write amplification* is defined as $b/(b-j)$. In general, the write amplification of an SSD composed of N blocks is defined as

$$A^N = \lim_{t\to\infty} \frac{b}{b - \sum_{j=1}^{b} j p_j(t)},$$

[1] In practice one avoids the need to copy the valid pages to RAM by making use of a single *free* block [7]

where $p_j(t)$ is the probability that the GC algorithm selects a block with j valid pages at time t, provided that the limit exists.

Denote the user-visible storage capacity as U blocks, i.e., bU pages, meaning the device *utilization* $\rho = U/N$ and the *spare factor* $S_f = 1 - U/N = 1 - \rho$. The objective of this paper is to analyze the write amplification and the distribution of the number of valid pages in a block for a class of GC algorithms under *uniform random writes*. Under uniform random writes there is no spacial or temporal locality, meaning the logical page number of a write request follows a uniform random distribution and is independent of all other write requests. We further assume that exactly bU pages are marked as valid at all times. Unless the operating system and SSD both support a command similar to the ATA TRIM command, the latter assumption corresponds to assuming that the SSD contains exactly bU valid pages at time 0. The ATA TRIM command allows the file system to inform the SSD that it can mark some pages as invalid when a file is deleted. Without it the number of pages in the valid state remains equal to bU at all times and also becomes equal to bU after a while if the SSD was initially empty.

The above implies that the probability that an external write operation "updates" a page stored on a block with exactly i valid pages is proportional to i/bU times the number of blocks containing exactly i pages. Note that the balls in our balls and bins system are selected uniformly at random as we consider an SSD drive with a uniform random write workload. Read and sequential write operations result in a far lower write amplification, hence the performance of the GC algorithm under random writes is the most significant [14].

It is possible to extend the analysis presented in this paper to the hot/cold data model of Rosenblum [17]. In this model a fraction f of the complete address space corresponds to *hot* data and the remaining fraction to *cold* data. The fraction of write operations to the hot data is denoted as r. Typical case studies assume that $f \leq 0.2$ and $r \geq 0.8$, meaning more than 80% of the writes are to less than 20% of the data [7].

We do not consider the issue of wear leveling in our problem setting. Wear leveling mechanisms try to prolongate the lifetime of the SSD by making sure that the number of write-erase cycles on a block does not vary too much. Some static wear leveling algorithms simply swap entire blocks (basically to move cold data to more worn out blocks), for instance by swapping the least and most worn out block or by swapping the free block with a randomly selected block as in Ban's algorithm [3]. When this type of swapping is used, the distribution of the number of valid pages is not affected by the wear leveling algorithm.

3 Related Work

Most of the analytic studies on GC algorithms have focused on the following three algorithms:

1. The Greedy GC algorithm selects a block that contains the least number of valid pages among all the blocks.
2. The FIFO GC algorithm selects the least-recently-written block, that is, the blocks are selected in a circular manner.
3. The Windowed GC algorithm makes use of a window of size $w \in \{1,\ldots,N\}$. It selects the block with the least number of valid pages among the set of the w least-recently-written blocks.

A highly accurate approximation for the write amplification of the Greedy algorithm under uniform random writes in a system where the number of blocks N and pages per block b is large, was introduced in [13,16] and can be expressed as

$$A^N \approx \frac{1}{1+\rho W(-e^{-1/\rho}/\rho)},$$

where $W(\cdot)$ is LambertW function (i.e., the inverse of $f(x) = xe^x$). This formula was also rediscovered in [19] and a less accurate approximation was also proposed in [1]. The above expression for A is also highly accurate for the write amplification of the FIFO algorithm [7] for large N, meaning the write amplification of the FIFO algorithm is independent of the block size b and coincides with the Greedy algorithm if b is large. The distribution of the number of valid pages per block and the write amplification of the Greedy algorithm for arbitrary b values (and large N) was analyzed in [5] and [7]. An analytic model for the write amplification of the Windowed GC algorithm was introduced in [10], but tends to result in an optimistic estimate of the write amplification [5,7]. The write amplification of the FIFO and Greedy GC algorithm with hot/cold data was also analyzed in [7], though the FIFO GC algorithm is not very suitable in the presence of hot and cold data as it also selects all the blocks containing lots of cold data.

4 A class of GC algorithms

In this paper we introduce a mean field model to assess the write amplification and distribution of the number of valid pages in a block for a class $\mathcal{C}$ of GC algorithms defined as follows. A GC algorithm belongs to class $\mathcal{C}$ if and only if the following two conditions hold:

C1: Let m_i be the fraction of blocks containing exactly i valid pages and denote $\mathbf{m} = (m_0,\ldots,m_b)$, then there should exist a set of probabilities $p_j(\mathbf{m})$ where $p_j(\mathbf{m})$ reflects the probability that a block containing exactly j valid pages is selected by the GC algorithm. In other words, whether block n, for any n, is selected by the GC algorithm should only depend on the number of valid pages j on block n and the fraction of blocks m_i containing exactly i valid blocks, for $i = 0,\ldots,b$.

C2: For $j = 0,\ldots,b$, the probabilities $p_j(\mathbf{m})$ should be smooth in $\mathbf{m}$ with $\mathbf{m} \in \Delta = \{\mathbf{m} \in \mathbb{R}^{b+1} | 0 \leq m_i \leq 1, \sum_{i=0}^b m_i = 1, \sum_{i=1}^b i m_i = b\rho\}$.

The following algorithms belong to class $\mathcal{C}$, where to the best of our knowledge the RANDOM++ and d-CHOICES GC algorithm have not been proposed before as a GC algorithm:

1. The RANDOM GC algorithm simply selects a block uniformly at random, hence $p_j(\mathbf{m}) = m_j$. The RANDOM+ algorithm operates in the same manner, except that it repeatedly selects another block as long as the selected block contains b valid pages (as it is useless to erase a full block). We therefore have $p_j(\mathbf{m}) = m_j/(1-m_b)$, which is well defined in Δ for $\rho < 1$.
2. The RANDOM++ GC algorithm repeatedly selects another block uniformly at random until it finds a block with at most $\lfloor b\rho \rfloor$ valid pages, hence
$$p_j(\mathbf{m}) = \frac{m_j 1[j \leq \lfloor b\rho \rfloor]}{\sum_{\ell=0}^{\lfloor b\rho \rfloor} m_\ell}, \tag{1}$$
where $1[A] = 1$ if A is true and 0 otherwise, which is also well-defined in Δ.
3. The d-CHOICES GC algorithm selects $d \geq 2$ blocks uniformly at random and erases a block containing the least number of valid pages among the d selected blocks. As all the selected pages must contain at least j valid pages, but not $j+1$, we have
$$p_j(\mathbf{m}) = \left(\sum_{\ell=j}^{b} m_\ell\right)^d - \left(\sum_{\ell=j+1}^{b} m_\ell\right)^d. \tag{2}$$

The write amplification of the RANDOM GC algorithm is clearly equal to $1/(1-\rho)$ as a block contains $b\rho$ valid pages on average. In this paper we will provide an explicit expression for the distribution of the number of valid pages in a block under the RANDOM algorithm as N, the number of blocks, tends to infinity. The write amplification of the RANDOM+ algorithm is less obvious to analyze and we will prove that it converges to $A = \frac{b}{b-\rho(b-1)}$ as N tends to infinity. We will also provide closed form expressions for the write amplification and distribution of the number of valid pages in a block for the RANDOM++ algorithm as N tends to infinity, while for the d-CHOICES algorithm we propose a fast numerical method to determine these performance measures using a set of ODEs. For the latter two results some open issues remain in order to formally prove that the obtained write amplification coincides with the limit of A^N as N tends to infinity (see Section 5.2). Similar to the RANDOM+(+) algorithm we can also define a d-CHOICES+(+) algorithm, however as soon as d exceeds 10 it is not very likely that the block with the least number of valid pages contains more than $\lfloor b\rho \rfloor$ valid pages; hence, the difference with the performance of the d-CHOICES algorithm is rather limited.

5 Mean field model

5.1 Model definition

We define a discrete-time system by observing the system state at the time epochs just prior to the operation of the GC algorithm. Hence, in between two observations the following steps take place:

S1: The GC algorithm selects a block as the new write frontier, say block number i, and copies the j valid pages of block number i to RAM.
S2: Block number i is erased and the j valid pages are copied back from RAM to the first j pages of the new write frontier, leaving the remaining $b-j$ pages in the erase state.
S3: The pages of the next $b-j$ random writes are invalidated and written to the remaining $b-j$ pages of the write frontier.

To analyze the performance of a GC algorithm belonging to class $\mathcal{C}$, we rely on the interacting objects framework introduced in [4]. Assume the device consists of N blocks, labeled 1 to N, that each store b pages (the state of which is erase, valid or invalid).

Let $X_n^N(t) \in S = \{0, 1, \ldots, b\}$, for $n = 1, \ldots, N$, be the number of valid pages on block number n at time t (i.e., when the GC algorithm runs for the t-th time). Let $M^N(t)$ be the occupancy measure of $X_n^N(t)$, that is, $M^N(t) = (M_0^N(t), M_1^N(t), \ldots, M_b^N(t))$ and

$$M_i^N(t) = \frac{1}{N} \sum_{n=1}^{N} 1[X_n^N(t) = i],$$

for $i = 0, \ldots, b$. Define

$$P_{i,i'}^N(\mathbf{m}) = \mathbb{P}[X_n^N(t+1) = i' | X_n^N(t) = i, M^N(t) = \mathbf{m}],$$

for $i \neq i' \in S$, that is, it contains the probability that the number of valid pages on block number n changes from i to i' during a single transition given the occupancy measure.

Define the set $\Delta^N = \{\mathbf{m} \in \mathbb{R}^{b+1} | m_i N \in \{0, 1, \ldots, N\}, i \in S, \sum_{i \in S} m_i = 1, \sum_{i \in S} i m_i = b\rho\}$ and let $p_j(\mathbf{m})$, for $j \in S$, be the probability that the GC selects a block with j valid pages at time t provided that $M^N(t) = \mathbf{m}$ with $\mathbf{m} \in \Delta^N$. To simplify the notation we also define the binomial probabilities $B_j(n, p) = \binom{n}{j} p^j (1-p)^{n-j}$.

To determine $P_{i,i'}^N(\mathbf{m})$, we note that the number of valid pages of block number n only changes if the block is selected during step $S1$ or if at least one of the random write operations in during step $S3$ involves block number n. Hence, the number of valid pages of at most $b+1$ blocks changes during a single transition. As explained below, this results in

$$\begin{aligned}
P^N_{i,i'}(\mathbf{m}) = \frac{p_i(\mathbf{m})}{m_i N} B_0(b-i, i/b\rho N) 1[i' = b] + \\
1[i' = i-1] \left[\sum_{j=1, j \neq b-i}^{b} p_{b-j}(\mathbf{m}) B_1(j, i/b\rho N) + \right. \\
\left. p_i(\mathbf{m}) \left(1 - \frac{1}{m_i N}\right) B_1(b-i, i/b\rho N) \right] + o(1/N), \qquad (3)
\end{aligned}$$

for $i \neq i' \in S$ and $m_i > 0$. Note, $p_i(\mathbf{m})/(m_i N)$ is the probability that the GC algorithm selects block n provided that it contains i valid pages, while $i/(b\rho N)$ is the probability that block number n is involved in a random write operation provided that it contains i valid pages. In other words, the first term corresponds to the case where block n is selected by the GC algorithm, while none of the $b-i$ writes involve block n, which implies that block n contains b valid pages at time $t+1$. The second and third term corresponds to the case where the GC algorithm does not select block number n, while exactly one of the random write operations in step $S3$ invalidates one of the i pages of block number n and therefore decreases its number of valid pages by one. Finally, all the other cases, where either (a) block n is involved in two or more write operations or (b) where block number n is selected by the GC algorithm and is involved in at least one random write operation, are covered by the $o(1/N)$ term as they are of the form $1/N^k$ with $k \geq 2$.

When $m_i = 0$ we can define $P^N_{i,i'}(\mathbf{m})$ as in (3) except that the terms $\frac{p_i(\mathbf{m})}{m_i}$ need to be replaced by the partial derivative $\partial p_i(\mathbf{m})/\partial m_i$, which is properly defined as $p_j(\mathbf{m})$ is smooth in Δ.

Define the drift $\mathbf{f}^N(\mathbf{m})$ for $\mathbf{m} \in \Delta^N$ as the expected change to M^N in one transition, that is,

$$\begin{aligned}
\mathbf{f}^N(\mathbf{m}) &= \mathbb{E}[M^N(t+1) - M^N(t) | M^N(t) = \mathbf{m}] \\
&= \sum_{i \neq i' \in S} m_i \mathbb{P}^N_{i,i'}(\mathbf{m})(e_{i'} - e_i), \qquad (4)
\end{aligned}$$

where e_i is the $(i+1)$-th row of the identity matrix of size $b+1$. Let $\mathbf{f}^N(\mathbf{m}) = (f^N_0(\mathbf{m}), \ldots, f^N_b(\mathbf{m}))$, then combining (3) and (4) yields

$$\begin{aligned}
f^N_b(\mathbf{m}) = \sum_{i=0}^{b-1} \frac{p_i(\mathbf{m})}{N} B_0(b-i, i/b\rho N) \\
- m_b \sum_{j=1}^{b} p_{b-j}(\mathbf{m}) B_1(j, 1/\rho N) + o(1/N), \qquad (5)
\end{aligned}$$

which is also valid for $m_i = 0$. The first term corresponds to the case where $i < b$ and $i' = b$, while for the second term $i = b$ and $i' = b-1$. For $i < b$, (3) and (4) result in

$$\begin{aligned}
f_i^N(\mathbf{m}) &= -\frac{p_i(\mathbf{m})}{N} B_0(b-i, i/b\rho N) \\
&+ m_{i+1} \sum_{j=1, j\neq b-(i+1)}^{b} p_{b-j}(\mathbf{m}) B_1(j, (i+1)/b\rho N) \\
&+ p_{i+1}(\mathbf{m}) \left(m_{i+1} - \frac{1}{N} \right) B_1(b-(i+1), (i+1)/b\rho N) \\
&- m_i \sum_{j=1, j\neq b-i}^{b} p_{b-j}(\mathbf{m}) B_1(j, i/b\rho N) \\
&- p_i(\mathbf{m}) \left(m_i - \frac{1}{N} \right) B_1(b-i, i/b\rho N) + o(1/N), \qquad (6)
\end{aligned}$$

which is also valid for $m_i = 0$.

Next, define the intensity function $\epsilon(N) = 1/N$ and let

$$\begin{aligned}
P_{i,i'}(\mathbf{m}) &= \lim_{N\to\infty} \frac{P_{i,i'}^N(\mathbf{m})}{\epsilon(N)} \\
&= \frac{p_i(\mathbf{m})}{m_i} 1[i' = b] + \left(\sum_{j=1}^{b} p_{b-j}(\mathbf{m}) j \right) \frac{i}{b\rho} 1[i' = i-1], \qquad (7)
\end{aligned}$$

for $m_i > 0$ due to (3). For $m_i = 0$ it is again sufficient to replace $\frac{p_i(\mathbf{m})}{m_i}$ by $\partial p_i(\mathbf{m})/\partial m_i$.

Similarly define $\mathbf{f}(\mathbf{m}) = (f_0(\mathbf{m}), \ldots, f_b(\mathbf{m}))$ such that for $i \in S$, $f_i(\mathbf{m}) = \lim_{N\to\infty} \frac{f_i^N(\mathbf{m})}{\epsilon(N)}$, then due to (5) and by noting that $\sum_{i=0}^{b-1} p_i(\mathbf{m}) = 1 - p_b(\mathbf{m})$, we find

$$f_b(\mathbf{m}) = (1 - p_b(\mathbf{m})) - \left(\sum_{j=1}^{b} p_{b-j}(\mathbf{m}) j \right) \frac{bm_b}{b\rho}, \qquad (8)$$

while for $i < b$, (6) yields

$$f_i(\mathbf{m}) = \frac{(i+1)m_{i+1} - im_i}{b\rho} \left(\sum_{j=1}^{b} p_{b-j}(\mathbf{m}) j \right) - p_i(\mathbf{m}). \qquad (9)$$

Finally, as in [4] define $\bar{M}^N(\tau)$ as the re-scaled process such that $\bar{M}^N(t) = M^N(\lfloor tN \rfloor)$, for $t \geq 0$. Similarly, define $\bar{X}_n^N(t)$ as the re-scaled version of $X_n^N(t)$. Further, define the deterministic process $\boldsymbol{\mu}(t) = (\mu_0(t), \ldots, \mu_b(t))$, the evolution of which is given by the following ODE:

$$\frac{d\boldsymbol{\mu}(t)}{dt} = \mathbf{f}(\boldsymbol{\mu}(t)), \qquad (10)$$

where $\mathbf{f}(\mathbf{m}) = (f_0(\mathbf{m}), \ldots, f_b(\mathbf{m}))$ is defined by (8) and (9).

5.2 Convergence result

From the previous section, $\{(X_1^N(t), \dots, X_N^N(t)), t \in \mathbb{N}\}$ is clearly a Markov chain on the state space Δ^N. A key feature of this Markov chain is that the state changes of X_n^N, for $n = 1, \dots, N$, are given by the probabilities $P_{i,i'}^N(\mathbf{m})$, meaning the evolution of X_n^N depends on X_k^N, with $k \neq n$, only through the occupancy measure $M^N(t)$.

The mean field interaction model in [4] considers a more general class of Markov chains $\{(X_1^N(t), \dots, X_N^N(t), R^N(t)),\ t \in \mathbb{N}\}$ with state space $\Delta^N \times \{1, \dots, J\}$. $R^N(t)$ is the state of the so-called resource at time t and the evolution of X_n^N depends on the occupancy measure $M^N(t)$ and the state $R^N(t)$. Further, the model is said to use no resource if $J = 1$, meaning $R^N(t)$ is a single state Markov chain.

The convergence results presented in [4] hold if five conditions, called Conditions $H1$ to $H5$, are satisfied. Conditions $H1$ and $H4$ are related to the resource and hold trivially for $J = 1$. Condition $H2$ demands that there exists a function $\epsilon(N)$, with $\lim_{N\to\infty} \epsilon(N) = 0$, and the limits $\mathbf{f}(\mathbf{m}) = \lim_{N\to\infty} \mathbf{f}^N(\mathbf{m})/\epsilon(N)$, given by (8) and (9) in our model, are properly defined. In fact the stronger condition $H2a$, which demands that $P_{i,i'}(\mathbf{m}) = \lim_{N\to\infty} P_{i,i'}^N(\mathbf{m})/\epsilon(N)$ is well defined, holds in our case as it is given by (7).

Given that $H2a$ holds, condition $H3$ demands that the coefficient of variation of the number of objects that change their state in a single transition is bounded for large N. As at most $b + 1$ objects can change their state in a single transition condition $H3$ is satisfied. Finally, condition $H5$ demands that $\mathbf{f}^N(\mathbf{m})$, given by (5) and (6) in our model, is a smooth function of $\mathbf{m}$ and $1/N$. This condition is met as $\mathbf{f}^N(\mathbf{m})$ is a polynomial function of $1/N$ (this is also true for the $o(1/N)$ term) and $p_j(\mathbf{m})$ is smooth in Δ. The following theorem therefore follows from Corollary 1 in [4].

Theorem 1 *If $M^N(0) \to \mathbf{m}$ in probability as N tends to infinity, then $\sup_{0\leq\tau\leq T} ||\bar{M}^N(t) - \boldsymbol{\mu}(t)|| \to 0$ in probability, where $\boldsymbol{\mu}(t)$ is the unique solution of the ODE* (10) *with $\boldsymbol{\mu}(0) = \mathbf{m}$.*

In other words, for N large and finite t, we can approximate $M^N(t)$ by $\boldsymbol{\mu}(t/N)$, which is the unique solution of the ODE (10) with $\boldsymbol{\mu}(0) = M^N(0)$. As we are interested in the stationary regime of $M^N(t)$, the question remains whether the convergence extends to the stationary regime. Corollary 2 in [4] shows that it suffices to show that the ODE given by (10) has a unique fixed point that is also a global attractor.

For the Random(+) GC algorithm, we provide an explicit expression for the unique fixed point of the ODE given by (10) and prove global attraction. For the Random++ algorithm we have an explicit expression for the unique fixed point (but no proof of global attraction), while for the d-Choices algorithm, we have no closed form results for the fixed point and only a proof of a unique global attractor for $b = 2$. Instead we numerically determine a fixed point of (10) and show by means of simulation that it is highly accurate in predicting the write amplification of the d-Choices GC algorithm.

6 Analytic results

In this section we study the set of ODEs given by (10) in more detail for some GC algorithms belonging to class $\mathcal{C}$.

6.1 The Random(+) GC algorithm

In this subsection we consider the Random GC algorithm. In this particular case $p_j(\mathbf{m}) = m_j$ and

$$\sum_{j=1}^{b} m_{b-j} j = b - \sum_{j=0}^{b} m_{b-j}(b-j) = (1-\rho)b,$$

for $\mathbf{m} \in \Delta$. As a result (8) reduces to

$$f_b(\mathbf{m}) = (1 - m_b) - \frac{1-\rho}{\rho} b m_b, \tag{11}$$

while for $i < b$, (9) yields

$$f_i(\mathbf{m}) = \frac{1-\rho}{\rho}[(i+1)m_{i+1} - i m_i] - m_i, \tag{12}$$

From (11) it follows that $\mu_b = \rho/(\rho + (1-\rho)b)$ for any fixed point $\boldsymbol{\mu} = (\mu_0, \ldots, \mu_b)$, while (12) implies that $\mu_i = \mu_{i+1}(1-\rho)(i+1)/(\rho + (1-\rho)i)$ holds, for $i = 0, \ldots, b-1$. Hence, we may conclude that (10) has a unique fixed point given by

$$\mu_i = \frac{\rho}{\rho + (1-\rho)i} \prod_{j=i+1}^{b} \frac{(1-\rho)j}{\rho + (1-\rho)j}, \tag{13}$$

for $i = 0, \ldots, b$. To prove global attraction of the unique fixed point $\boldsymbol{\mu}$, we note that (10) can be written as

$$\frac{d\boldsymbol{\mu}(t)}{dt} = e_b + \boldsymbol{\mu}(t) \underbrace{\begin{bmatrix} -1 & & & \\ \frac{1-\rho}{\rho} & -(1+\frac{1-\rho}{\rho}) & & \\ & \ddots & \ddots & \\ & & \frac{(1-\rho)b}{\rho} & -(1+\frac{(1-\rho)b}{\rho}) \end{bmatrix}}_{\text{matrix } Q}. \tag{14}$$

Hence, the unique solution $\boldsymbol{\mu}(t)$ is given by

$$\boldsymbol{\mu}(t) = e_b(-Q)^{-1}(I - e^{tQ}) + \boldsymbol{\mu}(0)e^{tQ},$$

and $\lim_{t\to\infty} \boldsymbol{\mu}(t) = e_b(-Q)^{-1} = \boldsymbol{\mu}$, for any $\boldsymbol{\mu}(0) \in \Delta$, as the diagonal entries of the bidiagonal matrix Q are negative and therefore $\lim_{t\to\infty} e^{tQ} = 0$.

Theorem 2 *Let μ_i^N be the steady state probability that an arbitrary block contains i valid pages when the* Random *GC algorithm is used in a system composed of N blocks of size b and spare factor S_f then*

$$\lim_{N\to\infty} \mu_i^N = \mu_i = \frac{\rho}{\rho+(1-\rho)i} \prod_{j=i+1}^{b} \frac{(1-\rho)j}{\rho+(1-\rho)j}, \tag{15}$$

for $i = 0,\dots,b$, where $\rho = 1 - S_f$. Further, let $w_i = \sum_{k=i}^b \mu_k$, then $w_0 = 1$ and

$$w_i = 1 - \prod_{j=i}^{b} \frac{(1-\rho)j}{\rho+(1-\rho)j}, \tag{16}$$

for $i = 1,\dots,b$. Finally, $\sum_{i=1}^b w_i = b\rho$.

Proof As noted in Section 5.2, the limit in (15) now follows from Corollary 2 of [4]. To establish the relationship for w_i, for $i = 1,\dots,b$, we first note that μ_i can also be written as

$$\mu_i = \frac{\left(\prod_{j=1}^{i-1}(\rho+(1-\rho)j)\right)\rho^{1[i>0]}\left(\prod_{j=i+1}^{b}(1-\rho)j\right)}{\prod_{j=1}^{b}(\rho+(1-\rho)j)},$$

which also confirms that $\sum_{i=0}^b \mu_i = 1$. Hence, for $i = 1,\dots,b$,

$$w_i = \frac{\prod_{j=i}^{b}(\rho+(1-\rho)j) - \prod_{j=i}^{b}(1-\rho)j}{\prod_{j=i}^{b}(\rho+(1-\rho)j)}.$$

Finally, using (16), we note that $\sum_{i=1}^b w_i = b\rho$ if and only if

$$\sum_{i=1}^{b}\left(\prod_{j=1}^{i-1}(\rho+(1-\rho)j)\right)\left(\prod_{j=i}^{b}(1-\rho)j\right) = \\ (1-\rho)b\prod_{j=1}^{b}(\rho+(1-\rho)j),$$

which can be proven easily by induction on b (starting with $b = 1$).

Theorem 2 confirms that the write amplification $A = \lim_{N\to\infty} A^N = b/(b - \sum_{i=1}^b w_i) = 1/(1-\rho)$, as noted in Section 4. The write amplification is thus independent of the block size b and the number of blocks N when the Random GC algorithm is used. The distribution of the number of valid pages within a block does however depend on both b and N. Theorem 2 provides a closed form expression for this distribution as N tends to infinity. To the best of our knowledge this concerns a new result that also enables us to determine the write amplification of the Random+ algorithm.

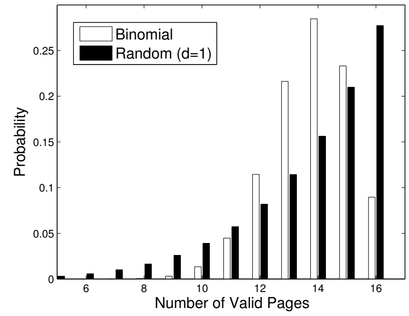

Figure 1 Distribution of the number of valid pages within a block for $S_f = (1-\rho) = 0.14$ and $b = 16$, compared to the binomial distribution with parameters (b, ρ).

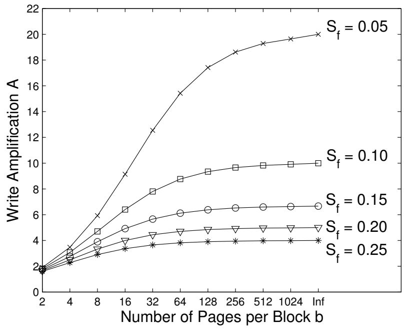

Figure 2 The write amplification A of the Random+ GC algorithm as a function of the block size b for different spare factors $S_f = 1 - \rho$.

Figure 1 depicts the distribution of the number of valid pages within a block for $b = 16$ and $\rho = 0.86$ compared to the Binomial distribution with parameters (b, ρ). The figure shows that the distribution of the number of valid pages is not close to Binomial as is sometimes assumed when analyzing GC algorithms. Thus, pages belonging to different blocks become independent for large N (due to the decoupling), but this is not the case for pages part of the same block as this would result in a Binomial distribution.

We end this section by considering the write amplification A of the Random+ algorithm, which operates similar to the Random GC algorithm, except that it repeatedly selects another block at random if the selected block contains b valid pages. The distribution of the number of valid pages per block is clearly identical for the Random and Random+ algorithm (this can also be seen from (8) and (9)). The expression for the write amplification however changes from $A = b/(b - \sum_{i=0}^{b} i\mu_i) = 1/(1-\rho)$ for the Random algorithm to

$$A = \frac{b}{b - \sum_{i=0}^{b-1} i \frac{\mu_i}{1-\mu_b}} = \frac{b}{b - \frac{b\rho - b\mu_b}{1-\mu_b}},$$

for the Random+ algorithm, which results in the following Corollary.

Corollary 1 *Let A^N be the write amplification of the* Random+ *algorithm in a system composed of N blocks of size b and spare factor $S_f = 1 - \rho$ then*

$$\lim_{N\to\infty} A^N = \frac{b}{b - \rho(b-1)}.$$

It shows that A is no longer independent of the block size b and that as b tends to infinity the Random and Random+ algorithm perform alike (as expected). We also note that the write amplification of the Random+ algorithm is bounded above by b irrespective of the spare factor S_f. Figure 2 depicts the write amplification A of the Random+ algorithm as a function of b for different values of $\rho = 1 - S_f$.

6.2 The d-Choices GC algorithm

In this subsection we consider the d-Choices GC algorithm with $d > 1$. Using (2) we can write

$$\sum_{j=1}^{b} p_{b-j}(\mathbf{m})j = b - \sum_{j=1}^{b}\left(\sum_{k=j}^{b} m_k\right)^d. \tag{17}$$

Let $\mu(t) = (\mu_0(t), \ldots, \mu_b(t))$ be the unique solution of (10) with initial condition $\mu(0)$. Define $w_i(t) = \sum_{k=i}^{b} \mu_k(t)$, for $i = 0, \ldots, b$, and $w_{b+1}(t) = 0$. Then, by means of (8) and (9), we find $w_0(t) = 1$ and

$$\frac{dw_i(t)}{dt} = 1 - w_i(t)^d - \left(b - \sum_{j=1}^{b} w_j(t)^d\right)\frac{i(w_i(t) - w_{i+1}(t))}{b\rho}, \tag{18}$$

for $i = 1, \ldots, b$.

Unless $d = 1$ (see Section 6.1), the set of equations given by (18) does not appear to have a simple closed form solution for its fixed point (for $d = b = 2$ we managed to obtain a closed form expression that already looks very involved). It is also unclear whether (10) has a global attractor in Δ, meaning we have no formal proof that the convergence to the mean field over finite time scales extends to the stationary regime for $d > 1$. When $b = 2$ the space Δ is one dimensional as $w_1(t) + w_2(t) = 2\rho$ and we can prove that a global attractor exists in Δ for any d (see Appendix A). Numerical experiments seem to suggest that a unique global attractor also exists for $b > 2$ and that the L_1-distance to the fixed point decreases along all the trajectories, as illustrated in Figure 3 for $b = 3$, $d = 4$ and $\rho = 0.75$.

To generate numerical results for the write amplification A and distribution of the number of valid pages for arbitrary b and ρ, we numerically solve the ODE given by (18) with $\mu_i(0) = \binom{b}{i}\rho^i(1-\rho)^{b-i}$ using Euler's method with a step size $h = 0.001$ until $||w(t+h) - w(t)||_1 < 10^{-13}$. For all the numerical experiments reported in this paper convergence occurred in a fraction of a

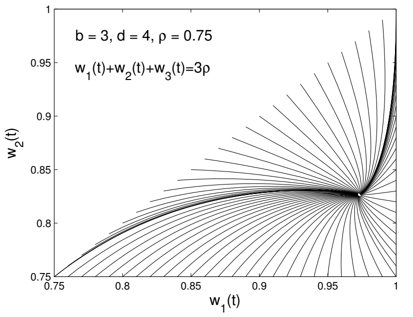

Figure 3 For $b = 3$ and $d = 4$ there is a unique global attractor in Δ for $\rho = 0.75$.

d	S_f	ODE (18)	simul. (95% conf.)
2	0.07	9.6354	9.6355 ±0.0016
4	0.07	7.7182	7.7181 ±0.0007
8	0.07	7.0044	7.0044 ±0.0004
2	0.14	4.9645	4.9651 ±0.0011
4	0.14	4.0672	4.0673 ±0.0008
8	0.14	3.7366	3.7366 ±0.0005
2	0.21	3.3732	3.3730 ±0.0006
4	0.21	2.8024	2.8026 ±0.0004
8	0.21	2.5936	2.5935 ±0.0002

Table 1 Comparison of ODE-based results and simulation experiments for a system with $N = 50,000$ blocks and $b = 64$ pages per block.

second. Tables 1 and 2 show a perfect agreement between the simulation results and the ODE-based prediction for a system consisting of $N = 50,000$ blocks[2] containing $b = 64$ and $b = 16$ pages, respectively. Depending on whether the page size is 4 or 8 Kilobyte, this results in a 12.8 or 25.6 Gigabyte system for $b = 64$. The simulation results in Tables 1 and 2 were based on 10 (for $S_f = 0.21$ and 0.14) and 50 (for $S_f = 0.07$) runs each with a length of $3tN$, where t is the smallest multiple of h such that $||w(t+h) - w(t)||_1 < 10^{-13}$. Initially the $b\rho N$ valid pages were distributed randomly over the Nb available pages and the length of the warm-up period was tN. As indicated in Tables 1 and 2 in each of the experiments the width of the 95% confidence intervals was smaller than 0.1%.

Remark The set of ODEs given by (18) has a simple intuitive explanation. As $1 - w_i(t)^d$ is the probability that the GC algorithm selects a block with less than i valid pages, it represents the rate at which blocks with i or more pages are created. Similarly, the rate at which blocks with i pages disappear is equal to $i(w_i(t) - w_{i+1}(t))/b\rho$, the probability that one of the write operations in step $S3$ involves a block with exactly i valid pages, times $b - \sum_{j=1}^{b} w_j(t)^d$, the mean number of writes between two executions of the GC algorithm.

[2] Similar results were obtained for a system consisting of $N = 5,000$ blocks.

d	S_f	ODE (18)	simul. (95% conf.)
2	0.07	8.9083	8.9078 ±0.0014
4	0.07	6.6296	6.6292 ±0.0010
8	0.07	5.7766	5.7766 ±0.0009
2	0.14	4.7339	4.7345 ±0.0020
4	0.14	3.7388	3.7383 ±0.0008
8	0.14	3.3612	3.3612 ±0.0007
2	0.21	3.2639	3.2636 ±0.0009
4	0.21	2.6480	2.6482 ±0.0004
8	0.21	2.4148	2.4149 ±0.0004

Table 2 Comparison of ODE-based results and simulation experiments for a system with $N = 50,000$ blocks and $b = 16$ pages per block.

If we let d tend to infinity in (18) the drift $f_i(\mathbf{w}(t)) = \frac{dw_i(t)}{dt}$ of the system satisfies the following equation:

$$f_i(w(t)) = 1[w_i(t) < 1] - \left(\sum_{j=1}^{b} 1[w_j(t) < 1]\right) \frac{i(w_i(t) - w_{i+1}(t))}{b\rho}. \tag{19}$$

As this drift function $\mathbf{f}$ is not smooth, we cannot rely on the framework presented in [4] for the Greedy algorithm. Instead, we can use the methodology developed in [8] to construct a differential inclusion (DI) from (19) as follows, such that the stochastic system converges to the solutions of the DI.

Let $\mathbf{y} = (1, \ldots, 1, y_{k+1}, \ldots, y_b)$, with $1 > y_{k+1} \geq \ldots \geq y_b \geq 0$. Define a set of vectors $\mathbf{u}_0(\mathbf{y}), \ldots, \mathbf{u}_k(\mathbf{y}) \in \mathbb{R}^b$ such that

$$\mathbf{u}_s(\mathbf{y}) = \lim_{y_{s+1},\ldots,y_k \to 1-} \mathbf{f}((\underbrace{1, \ldots, 1}_{s}, y_{s+1}, \ldots, y_k, y_{k+1}, \ldots, y_b)),$$

for $s = 0, \ldots, k$. Due to (19), we find

$$\mathbf{u}_s(\mathbf{y}) = (\underbrace{0, \ldots, 0}_{s}, 1, \ldots, 1) - \frac{(b-s)}{b\rho}(\underbrace{0, \ldots, 0}_{k-1}, k(1 - y_{k+1}), (k+1)(y_{k+1} - y_{k+2}), \ldots, b(y_b - y_{b+1})).$$

The set-valued function $\mathbf{F}(\mathbf{y})$ that characterizes the DI is then defined as the convex hull of $\mathbf{u}_0(\mathbf{y}), \ldots, \mathbf{u}_k(\mathbf{y})$. Thus, for any solution $\mathbf{w}(t)$ of the DI, with $\mathbf{w}(t) = (1, \ldots, 1, w_{k(t)+1}(t), \ldots, w_b(t))$ and $k(t) = \max\{i : w_i(t) = 1\}$, there exists an $\alpha_1(t), \ldots, \alpha_{k(t)+1}(t) \geq 0$, with $\sum_{i=1}^{k(t)+1} \alpha_i(t) = 1$, such that

$$\frac{dw_i(t)}{dt} = \sum_{j=1}^{k(t)+1} \alpha_j(t)(\mathbf{u}_{j-1}(\mathbf{w}(t)))_i,$$

that is,

$$\frac{dw_i(t)}{dt} = \tag{20}$$
$$\begin{cases} \sum_{j=1}^{i} \alpha_j(t) & i < k(t), \\ (1-\alpha_{k(t)+1}(t)) - \frac{k(t)}{b\rho}(1-w_{k(t)+1}(t)) & \\ \qquad \cdot \left(b - k(t) + \sum_{j=1}^{k(t)} (k(t)-j+1)\alpha_j(t)\right) & i = k(t), \\ 1 - \frac{i}{b\rho}(w_i(t) - w_{i+1}(t)) & \\ \qquad \cdot \left(b - k(t) + \sum_{j=1}^{k(t)} (k(t)-j+1)\alpha_j(t)\right) & i > k(t). \end{cases}$$

Let $\mathbf{w} = (1,\ldots,1,w_{k+1},\ldots,w_b)$ be a fixed point of the DI with $w_{k+1} < 1$ and $\alpha_1,\ldots,\alpha_{k+1}$ the corresponding convex combination of $\mathbf{u}_0(\mathbf{w}),\ldots,\mathbf{u}_k(\mathbf{w})$. Further denote $x_i = w_i - w_{i+1}$, for $i = 0,\ldots,b$, then (20) implies that $\alpha_1 = \ldots, \alpha_{k-1} = 0$ and

$$\alpha_k = (b-k+\alpha_k)\frac{k}{b\rho}x_k,$$
$$1 = (b-k+\alpha_k)\frac{i}{b\rho}x_i,$$

for $i > k$, which yields $x_k = \alpha_k \frac{b}{k} x_b$ and $x_i = \frac{b}{i} x_b$, for $i > k$ (while $x_i = 0$ for $i < k$). By noting that $\sum_{i=1}^{b} i x_i = b\rho$, we therefore have

$$x_b = \frac{\rho}{b-k+\alpha_k},$$

while $\sum_{i=0}^{b} x_i = 1$ implies

$$\alpha_k = \frac{k}{b\rho - k}\left[b - k - b\rho\left(\frac{1}{k+1} + \ldots + \frac{1}{b}\right)\right].$$

Finally, k is found as

$$k = \min\left\{i : b - i - b\rho\left(\frac{1}{i+1} + \ldots + \frac{1}{b}\right) > 0\right\},$$

as $\alpha_k \in (0,1]$. It is readily verified that the fixed point of the DI corresponds to the closed form expressions presented in [5] for the Greedy GC algorithm. It is also worth noting that α_k represents the probability that the GC algorithm selects a block containing exactly $k-1$ valid pages (note, the probability that an arbitrary block contains exactly $k-1$ valid pages is zero as $x_{k-1} = 0$), while $\alpha_{k+1} = 1 - \alpha_k$ represents the probability that the GC algorithm selects a block containing exactly k valid pages.

6.3 The Random++ GC algorithm

An expression for the probabilities $p_j(\mathbf{m})$ for the Random++ algorithm is given in (1), when combined with (8) and (9), this implies that any fixed point $\boldsymbol{\mu} = (\mu_0, \ldots, \mu_b)$ must fulfill the following set of equations

$$1 = \left(\sum_{j=1}^{b} p_{b-j}(\boldsymbol{\mu}) j \right) \frac{\mu_b}{\rho}, \tag{21}$$

$$i\mu_i = (i+1)\mu_{i+1}, \tag{22}$$

for $i = \lfloor b\rho \rfloor + 1, \ldots, b-1$ and

$$p_i(\boldsymbol{\mu}) = \frac{(i+1)\mu_{i+1} - i\mu_i}{b\rho} \left(\sum_{j=1}^{b} p_{b-j}(\boldsymbol{\mu}) j \right), \tag{23}$$

for $i = 0, \ldots, \lfloor b\rho \rfloor$. The following theorem shows that this set of equations has a unique solution in Δ.

Theorem 3 *The set of ODEs given by* (8) *and* (9) *for the* Random++ *GC algorithm, i.e., with* $p_j(\mathbf{m})$ *given by* (1)*, has a unique fixed point in* Δ *given by*

$$\mu_i = \frac{(i+1)\mu_{i+1}}{i + \rho/(1 - \rho - \mu_b(bS_{\rho,b} - b + \lfloor b\rho \rfloor))}, \tag{24}$$

for $i = 0, \ldots, \lfloor b\rho \rfloor$*, with* $S_{\rho,b} = \sum_{j=\lfloor b\rho \rfloor + 1}^{b} 1/j$*,*

$$\mu_i = b\mu_b/i, \tag{25}$$

for $i = \lfloor b\rho \rfloor + 1, \ldots, b-1$*, while*

$$\mu_b = \frac{-b_\rho + \sqrt{b_\rho^2 - 4a_\rho c_\rho}}{2a_\rho},$$

with $a_\rho = b - \lfloor b\rho \rfloor - bS_{\rho,b}$*,* $b_\rho = \rho S_{\rho,b} + 1 - \rho$ *and* $c_\rho = -\rho/b$ *for* $\rho < 1 - 1/b$ *and* $\mu_b = \rho/(\rho + (1-\rho)b)$ *for* $\rho \geq 1 - 1/b$*. Further,*

$$A \overset{def}{=} \frac{b}{\sum_{j=1}^{b} p_{b-j}(\boldsymbol{\mu}) j} = \frac{1}{1 - \frac{\rho - \mu_b(b - \lfloor b\rho \rfloor)}{1 - \mu_b b S_{\rho,b}}}. \tag{26}$$

Proof We start by noting that for $\boldsymbol{\mu} \in \Delta$

$$\sum_{j=1}^{b} p_{b-j}(\boldsymbol{\mu}) j = b - \sum_{j=1}^{b} p_j(\boldsymbol{\mu}) j =$$

$$b - \frac{\sum_{j=1}^{\lfloor b\rho \rfloor} j\mu_j}{1 - \sum_{j > \lfloor b\rho \rfloor} \mu_j} = b - \frac{b\rho - \sum_{j > \lfloor b\rho \rfloor} j\mu_j}{1 - \sum_{j > \lfloor b\rho \rfloor} \mu_j}.$$

S_f	Theorem 3	simul. (95% conf.)
0.20	2.9614	2.9611 ±0.0005
0.17	3.4209	3.4209 ±0.0004
0.14	4.0663	4.0663 ±0.0005
0.11	5.0371	5.0377 ±0.0007
0.08	6.6599	6.6601 ±0.0006
0.05	9.9172	9.9166 ±0.0010

Table 3 Comparison of closed form results and simulation experiments for a system with $N = 50,000$ blocks and $b = 32$ pages per block.

Due to (22), we have

$$\begin{aligned}\sum_{j>\lfloor b\rho\rfloor} j\mu_j &= b\mu_b(b-\lfloor b\rho\rfloor),\\ \sum_{j>\lfloor b\rho\rfloor} \mu_j &= b\mu_b S_{\rho,b}.\end{aligned} \tag{27}$$

This implies

$$\sum_{j=1}^{b} p_{b-j}(\boldsymbol{\mu})j = b\left(1-\frac{\rho-\mu_b(b-\lfloor b\rho\rfloor)}{1-\mu_b b S_{\rho,b}}\right),$$

which establishes (26), while (24) can now be derived from (23) and (25) is immediate from (22). The quadratic equation $f(y) = a_\rho y^2+b_\rho y+c_\rho = 0$ for μ_b now follows from (21). Provided that the function $f(y)$ has real roots, they are both positive as $a_\rho, c_\rho \leq 0$ and $b_\rho > 0$, while $\mu_b \leq 1/(\rho S_{\rho,b})$ as $\sum_{j>\lfloor b\rho\rfloor} \mu_j \leq 1$. Further,

$$f(0) < 0, \text{ and } f(1/(\rho S_{\rho,b})) = \frac{b-\lfloor b\rho\rfloor - b\rho S_{\rho,b}}{(bS_{\rho,b})^2}.$$

Hence, $f(1/(\rho S_{\rho,b})) \geq 0$ if and only if $b-\lfloor b\rho\rfloor - b\rho S_{\rho,b} \geq 0$. This latter inequality holds as $g(\rho) = b-\lfloor b\rho\rfloor - b\rho S_{\rho,b}$ is equal to $1-\rho$ for $\rho > 1-1/b$ and $g(\rho)$ increases as ρ decreases.

Provided that the unique fixed point is a global attractor, Theorem 3 implies that the write amplification A_N in a system consisting of N blocks converges to (26) as N tends to infinity. By means of (27) we also find that the mean number of attempts needed to locate a block with at most $\lfloor b\rho\rfloor$ valid blocks can be expressed as $1/(1-\mu_b b S_{\rho,b})$.

Table 3 compares the closed form expression for A given by Theorem 3 with simulation experiments on a system consisting of $N = 50,000$ blocks and $b = 32$ pages per block. The length of a single simulation run and warm-up period was determined in a similar manner as in Section 6.2, while 10 runs were performed for $S_f > 0.1$ and 50 for $S_f < 0.1$. The results show a perfect agreement between the closed form results and simulation.

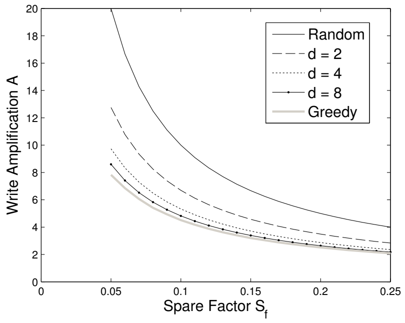

Figure 4 Write amplification A as a function of the spare factor S_f for the RANDOM, GREEDY and d-CHOICES algorithm for $d = 2, 4$ and 8 and $b = 32$ pages per block.

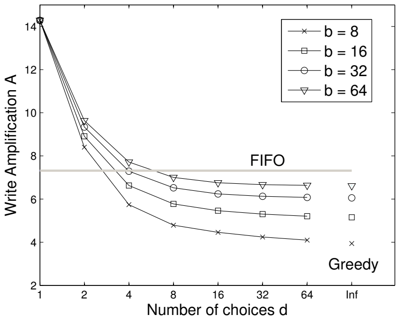

Figure 5 Write amplification A as a function of the number of choices d for the d-CHOICES algorithm with a spare factor $S_f = 0.07$.

7 Numerical results

In this section we present some numerical results for the d-CHOICES and RANDOM++ algorithm and compare their performance with the GREEDY, FIFO and WINDOWED algorithm.

7.1 The d-Choices GC algorithm

We will show that the d-CHOICES algorithm can approximate the write amplification of the GREEDY algorithm even for small d values, e.g., $d = 10$, while maintaining the simplicity of the RANDOM or FIFO algorithm. Further, we will show that the d-CHOICES algorithm is far more effective than the WINDOWED algorithm, that is, the d-CHOICES algorithm with d small, e.g., $d = 10$, has a lower the write amplification A than the WINDOWED algorithm with a fairly large window size, e.g., $w = 500$.

Figure 4 depicts the write amplification A as a function of the spare factor $S_f = 1 - \rho$ for the RANDOM, GREEDY and d-CHOICES GC algorithm for $d = 2, 4$ and 8 and $b = 32$ pages per block. The results for the write amplification

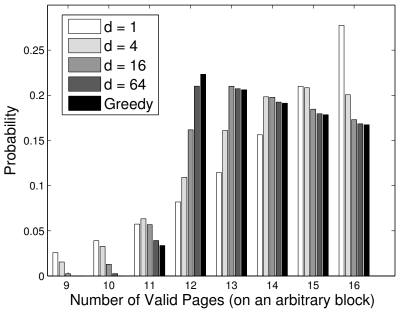

Figure 6 Distribution of the number of valid pages on an arbitrary block for the greedy and d-CHOICES algorithm with $d = 1, 4, 16$ and 64, with $S_f = 0.14$ and $b = 16$.

(and number of valid blocks) under the Greedy GC algorithm are based on [5]. The results confirm that a small value of d suffices to approximate the write amplification A of the Greedy algorithm, especially for larger spare factors S_f. Although the Greedy algorithm has a lower write amplification A, it requires state information (essentially $b + 1$ bins that contain N items in total) that needs to be updated after each write operation. The d-Choices GC algorithm maintains no state information and is only activated when a new block needs to be selected (and cleared).

In Figure 5 we also show the impact of the number of pages b per block on the write amplification A when the spare factor $S_f = 0.07$. It confirms that small d values suffice for the d-Choices algorithm to approximate the write amplification of the Greedy algorithm for different block sizes b. The FIFO algorithm, the write amplification of which does not depend on b, performs worse, especially for small b (i.e., older SSD devices) as the write amplification of the d-Choices and Greedy algorithm decreases with b (as expected).

When $b = 1$, meaning $N\rho$ blocks contain one valid page and $N(1-\rho)$ one invalid page at all times, the d-Choices GC algorithm has a write amplification $A = 1/(1-\rho^d)$ as with probability $1-\rho^d$ the selected block contains an invalid page. In fact for any $b \geq 1$, it is not hard to show that the write amplification of the d-CHOICES algorithm is lower bounded by $1/(1-\rho^d)$. This can be shown by noting that the write amplification $A(t)$ at time t is equal to $b/(b - \sum_{i=1}^b w_i(t)^d)$ and $\sum_{i=1}^b w_i^d$, for $d \geq 1$, is minimized in Δ when $w_i = \rho$ for $i = 1, \ldots, b$. We can also upper bound the write amplification A by

$$\frac{b}{b - \lfloor b\rho \rfloor - (b\rho - \lfloor b\rho \rfloor)^d},$$

by noting that $\sum_{i=1}^b w_i^d$, for $d \geq 1$, is maximized in Δ when $w_i = 1$ for $i = 1, \ldots, k$, $w_{k+1} = b\rho - k$ and $w_i = 0$ for $i = k+2, \ldots, b$ with $k = \lfloor b\rho \rfloor$. Note, when ρ is a multiple of $1/b$ this upper bound simplifies to $1/(1-\rho)$, the write amplification of the Random algorithm, otherwise the upper bound is below $1/(1-\rho)$ for $d > 1$.

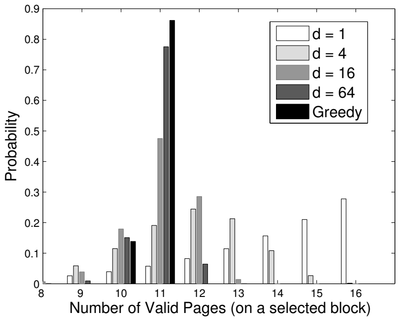

Figure 7 Distribution of the number of valid pages on a selected block for the greedy and d-CHOICES algorithm with $d = 1, 4, 16$ and 64, with $S_f = 0.14$ and $b = 16$.

The previous results indicated that the write amplification of the GREEDY and d-CHOICES algorithm becomes similar as d increases. Figures 6 and 7 indicate that the same holds for the number of valid pages in a block on an arbitrary and a block selected by the GC algorithm, respectively, for a system with $b = 16$ pages per block and a spare factor $S_f = 0.14$. Note, for the GREEDY algorithm the probability that an arbitrary block contains at most 10 valid pages is zero, while the number of valid pages on a selected block is bimodal and is always 10 or 11 in our example. Hence, at times a negligible fraction of the blocks contains exactly 10 pages and these blocks are always selected by the GREEDY GC algorithm [5,7]. For the d-CHOICES algorithm we observe something similar: the probability of having 10 valid pages in a block tends to zero as d increases, while the probability of selecting such a block remains significant. This can be understood by noting that even though such blocks become rare as d grows, larger d values also increase the probability that a rare block (containing the least number of valid pages) is selected by the GC algorithm.

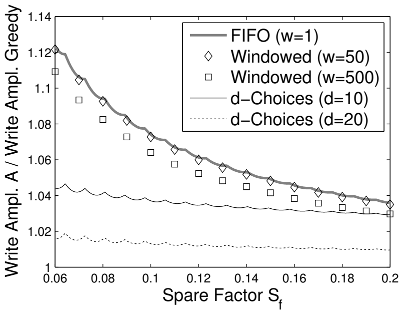

Figure 8 Relative write amplification WINDOWED versus d-CHOICES algorithm for $b = 64$ blocks per page.

The Windowed GC algorithm was introduced in [10] as a trade-off between the low complexity of the FIFO algorithm and the good performance of the Greedy algorithm. The idea is to consider only the w *oldest* blocks when searching for the block with the least number of valid pages, where setting $w = 1$ and N results in the FIFO and Greedy GC algorithm, respectively. Larger w values reduce the write amplification, but increase the time complexity of the GC algorithm. Figure 8 shows how much the write amplification increases when the Windowed (with $w = 50$ and 500) or d-Choices (with $d = 10$ or 20) algorithm is used instead of the Greedy algorithm in a system with $b = 64$ pages per block. Note, the curves in this figure are not smooth as the write amplification of the Greedy algorithm is not smooth in those S_f values for which the bimodal distribution of the number of valid pages on a selected block becomes unimodal.

Figure 8 indicates that for spare factors $S_f \leq 0.2$ setting d as small as 10 suffices to beat the Windowed algorithm with a window size of $w = 500$, where the gain becomes more pronounced as S_f decreases. Further, setting $d = 20$ results in a write amplification that is less than 2% above the write amplification of the Greedy algorithm, while the write amplification of the Windowed algorithm is still much closer to the FIFO algorithm even with a window size $w = 500$. This can be understood by remarking that blocks with a relatively high number of valid pages tend to stay within the window for a considerable amount of time. Such a drawback does not occur with the d-Choices algorithm as the set of d blocks is always reselected at random.

The fact that the Windowed GC algorithm is not very effective in reducing the write amplification for w small was also noted in [7]. The results in Figure 8 for the windowed access algorithm were obtained by simulation on a system with $N = 50,000$ blocks, using 10 runs of length 10^6 each. This resulted in confidence intervals with a width below 0.1%. Note, analytical results for the Windowed GC algorithm were also presented in [10], but these were based on the assumption that the number of valid pages per block within the window has a binomial distribution, which tends to result in an optimistic estimate for the write amplification [5,7].

7.2 The Random++ algorithm

In this section, we compare the write amplification of the Random++ algorithm with the FIFO and Greedy GC algorithm. We will show that the Random++ algorithm performs worse than the FIFO algorithm when the number of pages in a block is large, e.g., $b \geq 64$, while the reverse is mostly true for small block sizes, e.g., $b \leq 16$. We will also show that the Random++ algorithm typically requires less than three attempts to locate a block with at most $\lfloor b\rho \rfloor$ valid pages.

Figure 9 depicts the write amplification A of the FIFO, Greedy and Random++ GC algorithm as a function of the spare factors $S_f = 1-\rho$ for $b = 64$ pages per block. It shows that the Random++ algorithm is outperformed

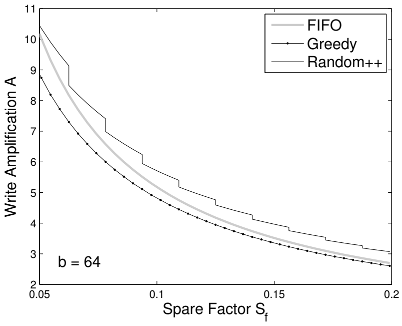

Figure 9 The write amplification A of the FIFO, Greedy and Random++ GC algorithm as a function of the spare factors $S_f = 1 - \rho$ for $b = 64$ pages per block.

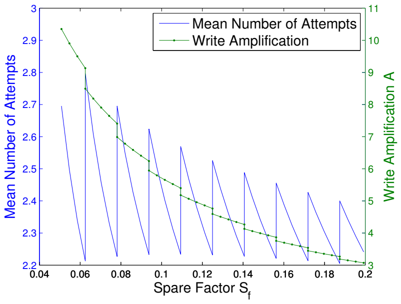

Figure 10 The write amplification A and mean number of attempts to find a block with at most $\lfloor b\rho \rfloor$ valid blocks for the Random++ GC algorithm as a function of the spare factors $S_f = 1 - \rho$ for $b = 64$ pages per block.

by the FIFO algorithm for $S_f \in [0.05, 0.2]$, especially when the spare factor becomes large. We also note that the curve of the Random++ algorithm contains jumps whenever the spare factor $S_f = 1 - \rho$ is a multiple of $1/b$. When S_f becomes a multiple of $1/b$ when increasing S_f, the maximum number of allowed valid pages in the block selected by the Random++ algorithm decreases by one. This causes an immediate decrease in the write amplification. At the same time we can also expect a sudden rise in the mean number of attempts needed by the Random++ GC algorithm to locate such a block as demonstrated in Figure 10. This figure also indicates that the mean number of attempts is between 2 and 3 for all $S_f \in [0.05, 0.2]$ for $b = 64$ pages per block.

Similar experiments, not depicted here, indicate that the Random++ GC algorithm does outperform the FIFO algorithm for $S_f \in [0.05, 0.2]$ when there are only $b = 8$ pages in a block. Whether the FIFO or Random++ algorithm achieves the lowest write amplification for $b = 16$ and 32 pages per block, depends in a complicated manner on the spare factor S_f (due to the jumps in the Random++ curve). We end by remarking that the write amplification A of the Random++ algorithm is well below that of the Random algorithm, the write amplification of which equals $1/(1-\rho)$, even for larger b values, e.g., $b = 64$.

8 Conclusions and future work

In this paper we introduced a mean field model to analyze the write amplification of a class $\mathcal{C}$ of garbage collection (GC) algorithms in flash-based solid state drives under uniform random writes. Algorithms belonging to class $\mathcal{C}$ include the Random(+), Random++ and d-Choices GC algorithms, where the latter two were analyzed for the first time. Closed form results for the write amplification and the distribution of the number of valid pages in a block were obtained for the Random(+) and Random++ algorithm, while a fast numerical ODE-based method was proposed for the d-Choices algorithm. The results were shown to be highly accurate using simulation experiments.

The d-Choices algorithm was shown to be very effective in reducing the write amplification, while the Random++ algorithm was less effective. More specifically, we showed that the d-Choices GC algorithm has a write amplification close to that of the Greedy GC algorithm even for small d values, e.g., $d = 10$, and offers a more attractive trade-off than the Windowed GC algorithm between its simplicity and its performance.

We are currently extending the mean field model for uniform random writes introduced in this paper, to the hot/cold data model of Rosenblum [17]. Preliminary results (not shown here) indicate that the write amplification of the d-Choices GC algorithm gets closer to the write amplification of the Greedy algorithm as the hot data gets hotter (i.e., as f decreases or r increases). In other words, even smaller d values suffice to get close to the performance of the Greedy GC algorithm.

We are also planning to extend the model to study the impact of data separation techniques for hot/cold data and of the TRIM command on the write amplification. The latter will make the model also more applicable to the setting of log-structured file systems where data is often deleted.

References

1. R. Agarwal and M. Marrow. A closed-form expression for write amplification in NAND flash. In *IEEE GLOBECOM Workshops (GC Wkshps)*, pages 1846–1850, 2010.
2. Y. Azar, A.Z. Broder, A.R. Karlin, and E. Upfal. Balanced allocations. In *SIAM Journal on Computing*, pages 593–602, 1994.
3. A. Ban. Wear leveling of static areas in flash memory. US patent 6,732,221. Filed June 1, 2001; Issued May 4, 2004; Assigned to M-Systems., 2004.
4. M. Benaïm and J. Le Boudec. A class of mean field interaction models for computer and communication systems. *Performance Evaluation*, 65(11-12):823–838, 2008.
5. W. Bux and I. Iliadis. Performance of greedy garbage collection in flash-based solid-state drives. *Perform. Eval.*, 67(11):1172–1186, November 2010.
6. F. Chen, D.A. Koufaty, and X. Zhang. Understanding intrinsic characteristics and system implications of flash memory based solid state drives. *ACM SIGMETRICS Perform. Eval. Rev.*, 37(1):181–192, 2009.

7. P. Desnoyers. Analytic modeling of SSD write performance. In *Proceedings of International Systems and Storage Conference (SYSTOR 2012)*, 2012.
8. N. Gast and B. Gaujal. Markov chains with discontinuous drifts have differential inclusion limits. *Perform. Eval.*, 69(12):623–642, 2012.
9. L. M. Grupp, J. D. Davis, and S. Swanson. The bleak future of NAND flash memory. In *Proc. of USENIX Conference on File and Storage Technologies*, 2012.
10. X. Hu, E. Eleftheriou, R. Haas, I. Iliadis, and R. Pletka. Write amplification analysis in flash-based solid state drives. In *Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference*, SYSTOR '09, pages 10:1–10:9, New York, NY, USA, 2009.
11. J.-U. Kang, H. Jo, J.-S. Kim, and J. Lee. A superblock-based flash translation layer for NAND flash memory. In *Proceedings of the 6th ACM & IEEE International conference on Embedded software*, EMSOFT '06, pages 161–170, New York, NY, USA, 2006.
12. Y. Li, P.P.C. Lee, and J.C.S. Lui. Stochastic modeling of large-scale solid-state storage systems: Analysis, design tradeoffs and optimization. *ACM SIGMETRICS Perform. Eval. Rev.*, 41(1), 2013.
13. J. Menon. A performance comparison of RAID-5 and log-structured arrays. In *Proceedings of the 4th IEEE International Symposium on High Performance Distributed Computing*, HPDC '95, pages 167–178, Washington, DC, USA, 1995.
14. C. Min, K. Kim, H. Cho, S. Lee, and Y. I. Eom. SFS: Random write considered harmful in solid state drives. In *Proc. of USENIX Conference on File and Storage Technologies*, pages 139–155, 2012.
15. M. Mitzenmacher, A. Richa, and R. Sitaraman. The power of two random choices: a survey of techniques and results. *Handbook of Randomized Computing*, 1, 2001.
16. J.T. Robinson. Analysis of steady-state segment storage utilizations in a log-structured file system with least-utilized segment cleaning. *SIGOPS Oper. Syst. Rev.*, 30(4):29–32, October 1996.
17. M. Rosenblum and J. K. Ousterhout. The design and implementation of a log-structured file system. *ACM Trans. Comput. Syst.*, 10(1):26–52, February 1992.
18. N.D. Vvedenskaya, R.L. Dobrushin, and F.I. Karpelevich. Queueing system with selection of the shortest of two queues: an asymptotic approach. *Problemy Peredachi Informatsii*, 32:15–27, 1996.
19. L. Xiang and B. Kurkoski. An improved analytical expression for write amplification in NAND flash. In *International Conference on Computing, Networking, and Communications (ICNC)*, pages 497–501, 2012.

A Uniqueness for b = 2

When $b = 2$, the space Δ is one dimensional and the evolution of $w_2(t)$ is given by

$$\frac{dw_2(t)}{dt} = 1 - w_2(t)^d - \left(2 - w_2(t)^d - (2\rho - w_2(t))^d\right) \frac{w_2(t)}{\rho},$$

due to (18) as $w_1(t) = 2\rho - w_2(t)$. Note, as $1 \geq w_1(t) \geq w_2(t) \geq 0$, $w_2(t) \in [\min(0, 2\rho-1), \rho]$. Define $g(w) = 1 - w^d - (b - w^d - (2\rho - w)^d)w/\rho$, then $g(\min(0, 2\rho - 1)) > 0$ and $g(\rho) = -(1 - \rho^d) < 0$. Further,

$$\begin{aligned} g'(w) = -dw^{d-1} - \left(2 - w^d - (2\rho - w)^d\right) \frac{1}{\rho} + \\ d\left(w^{d-1} - (2\rho - w)^{d-1}\right) \frac{w}{\rho}, \end{aligned}$$

meaning $g'(w) < 0$ for $w \in [\min(0, 2\rho - 1), \rho]$. Hence, there is a unique fixed point in Δ that is necessarily a global attractor.